Master's Thesis Nr. 50

Systems Group, Department of Computer Science, ETH Zurich

in collaboration with

Proteus Project, Courant Institute of Mathematical Sciences, New York University

Analysis of German Patent Literature

by

Luciano Franceschina

Supervised by

Prof. Donald Kossmann (ETH)
Prof. Ralph Grishman (NYU)

February 2012 - August 2012

Analysis of German Patent Literature

Luciano Franceschina
ETH Zürich
`lucianof@student.ethz.ch`

August 2012

Abstract

We show how several components of the JET natural language analysis tool, originally developed at New York University for the analysis of English text, were adapted to German. These components, such as the part of speech tagger and the noun chunker, are explained in terms that should be understandable to a layman. On the other hand, issues that arise specifically with regards to the German language are outlined in a way that could be of interest to people more experienced in natural language processing.

Acknowledgements

First, I would like to thank the members of the Proteus Project, and especially Prof. Ralph Grishman, for the opportunity to write my Master's thesis in this research group and the support they offered me during my stay at NYU.

I also want to thank Sandra Kübler (Department of Linguistics, Indiana University) for providing us with a chunk corpus derived from the German TüBa-D/Z treebank using the conversion strategy described in her paper [10] .

Research reported herein was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) of the U.S. Government via Department of Interior National Business Center contract number D11PC20154. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

Contents

Chapter 1

Introduction

In this report we describe the steps taken to adapt several components of a Java-based language analysis tool from English to German. In doing so we illustrate relevant linguistic differences between the two languages and discuss implementation details. Finally we will also present metrics and compare the performance of the German components to state-of-the-art systems. Specifically, components for part-of-speech tagging, chunking, as well as pattern and terminology extraction were adapted and developed for German.

The work was done within the context of a project to extract selected semantic relationships from the patent literature in English, German, and Chinese.

In the following two sections we introduce the tool and the project in greater detail and give an overview of this document.

1.1 Java Extraction Toolkit (JET)

The Java Extraction Toolkit (JET) is developed at NYU by members of the Proteus Project. JET includes components for a multitude of language analysis tasks - part-of-speech tagging, chunking, parsing, pattern application and more. These components can be accessed interactively via a GUI for the analysis of single sentences as well as in batch-mode, where several documents are analyzed in one run.

All components operate on the central Document class that contains text as well as a set of annotations of a document that are associated with a specific span of the document. The text is stored as an immutable string and annotations are accessed by their offset from the start of the string. The different components add and modify annotations, but never modify the text. Most components build upon each other and read annotations from lower level components. The first, basic component is the Tokenizer, which identifies separate words and annotates each word with a token annotation. Other components, such as the part of speech tagger (see chapter 2) or the chunker (see chapter 3) build upon this and add more information about the document structure in form of annotations. This architecture is based on the 'Tipster Architecture' [6].

1.2 Relation extraction from patents

As part of a larger effort at tracking developing technologies, we seek to extract some of the semantic relationships expressed in the text of English, German, and Chinese patents. The Jet system already provides a rich set of tools for English text analysis, mostly corpus-trained. We decided therefore to adapt some of these tools to German and train them using annotated German corpora. Specifically, we intend to identify instances of these relationships by matching a set of patterns (regular expressions) involving specific words, parts of speech, and some basic syntactic constituents ("chunks").

1.3 Overview

This document describes the components adapted and developed for the analysis of German patents. It follows the same order as the stages of language processing: in chapter 2 the part of speech tagger is introduced, in chapter 3 the chunker. Chapter 4 is a short intermezzo, in which we look at the challenges we have to deal with when analyzing patents. Finally, chapter 5 and 6 introduce the terminology extraction and Hearst pattern components, respectively.

Chapter 2

Part of speech tagging

Part of speech tagging is the cornerstone of most language analysis tasks. Ultimately, the goal of natural language processing is to get the machine to understand certain aspects of the meaning of a text. In order to do that, it is necessary to understand the building blocks of the text - the words. Identifying the part of speech of a word is the first step in this direction.

2.1 Hidden Markov Model

Identifying a word's part of speech would be trivial in a language with a finite set of words, where every word form can be exactly one part of speech. Human languages are not like that - the same word can belong to a different part of speech category depending on the context, and new words are formed frequently. Therefore, in order to assign a part of speech to a word we need a more complicated model of the language than just a direct mapping of word to part of speech. The part of speech tagger (POS tagger) of JET is based on a first-order Hidden Markov Model (HMM). A HMM takes two sets of probabilities into account: transition probabilities between hidden states and emission probabilities of observations given a state. For POS tagging, states refer to the part of speech tags and the observations are the words. We are therefore interested in these probabilities:

- The transition probabilities: The probability that one part of speech follows another one. For example, the sequence article$\rightarrow$noun is more likely to occur than article$\rightarrow$verb. We need the probability of each possible transition between two part of speech tags.
- The emission probabilities: The probability that given a part of speech, a specific word is emitted. For example the probability that given a certain

word is an article, this word is "the". We need the probability of every word for every part of speech tag.

Having these probabilities, and given a chain of words (i.e. a sentence) of which we don't know the parts of speech, it is possible to compute the most likely chain of hidden states (i.e. part of speech tags) that generated this observation. We assume that these are the parts of speech of the individual words of the sentence. For a more thorough description of HMMs and the Viterbi algorithm to compute the most likely sequence of hidden states we refer to [9].

2.2 Corpus selection

To obtain the necessary probabilities for our Hidden Markov Model, we train our software on a tagged corpus; a corpus where for every word, a human annotator has determined the correct part of speech tag. By counting the frequencies of the transitions ("tag follows tag") and emissions ("word is tagged as"), we can estimate the probabilities. The larger the corpus is, the better the estimate becomes. There are currently three large corpora for written German [15]:

- The TIGER corpus [1] of approximately 50,000 sentences taken from the German newspaper 'Frankfurter Rundschau' and developed by the Department of Computational Linguistics and Phonetics in Saarbrücken, the Institute of Natural Language Processing (IMS) in Stuttgart, and the Institut für Germanistik in Potsdam.
- The older NEGRA corpus [2] of approximately 20,000 sentences, also taken from the German newspaper 'Frankfurter Rundschau' and developed at the Saarland University in Saarbrücken and the IMS in Stuttgart.
- The TüBa-D/Z corpus [17] (Tübinger Baumbank des Deutschen / Zeitungskorpus) of approximately 65,000 newspaper sentences taken from 'die tageszeitung' (taz) and developed at the Seminar für Sprachwissenschaft at the Eberhard Karls University, Tübingen.

These corpora are also referred to as treebanks, because they are not just annotated with part of speech tags, but with the whole syntactic tree structure of the sentences. While the three treebanks slightly differ in their linguistic paradigms concerning the syntactic trees [11], they all use the same tag set for parts of speech - the Stuttgart-Tübingen-TagSet (STTS) [16], which consists of 54 different tags.

For this project, we decided to use the TüBa-D/Z corpus, as it is currently the largest one and is actively maintained: TüBa-D/Z Release 7, adding about 10'000 additional sentences to the treebank, was published in December 2011.

2.3 German and English parts-of-speech

The TüBa-D/Z corpus is tagged with the standard STTS tag set, which consists of 54 tag types (see Appendix A). JET's English POS tagger uses the Penn Treebank tag set [12] with 36 POS tag types, plus 13 additional tag types for punctuation and other special characters. Of the 54 tags in the STTS tag set, only three are for punctuation and one for non-words. The German set of proper part of speech tags is therefore considerably larger than the English one. For example, there are 12 tags for verbs in the German STTS, but only 6 in the Penn Treebank tag set.

2.4 Implementation details

In this section we discuss implementation details of the JET POS tagger in general, and adjustments that have been made for German POS tagging specifically.

2.4.1 Word features

By computing the emission probabilities purely from the observed frequencies in the training corpus, an unknown word (i.e. a word not present in the training corpus) would be assigned a probability of 0 for being emitted by any state - it would therefore not be possible to compute the most likely sequence of POS tags for sentences containing an unknown word, as all possible sequences would have probability 0. There are words that are only observed once in the training corpus too, though. The relative frequency of these singletons is an estimate of the probability for the observation of an unknown word. This gives us the probability for each part of speech tag to emit an unknown word. For example, an unknown word it is more probable to be a noun or an adjective than a pronoun.

In addition to this, the JET POS tagger also builds statistics on some word features, in order to get a better estimate for the emission probability of an unknown word. An unknown word's emission probability for each POS tag is estimated as the relative frequency of a singleton word times the relative frequency of the word's feature. The following word features are considered:

- *allDigits*: numbers
- *allCaps*: word in all capital letters
- *initCaps*: word with one initial capital letter
- *lowerCase*: word in all lowercase letters
- *hyphenated*: word contains a hyphen

- *other*: none of the above

For the German POS tagger, we added the following word features:

- *noAlphanumerics*: word without letters or numbers
- *digitsThenLetters*: word starts with digits and ends with letters
- *containsGe*: word contains "ge"

Each of these additional word features slightly improves the performance of the POS tagger. The feature *digitsThenLetters* is specifically useful for the recognition of certain age-related adjectives, such as "38jährig" (38 years old).

2.4.2 Affix statistics

German is a morphologically richer language than English. In German, verbs have up to 29 possibly different forms [4], while conjugated English regular verbs only take three possible suffixes (-s, -ed, -ing). German nouns, pronouns and adjectives are declined according to case, number and gender, while in English in most cases declension is only necessary for pluralization. We therefore have many more different word forms for a single word in German. This makes POS tagging harder, because we have fewer observations in our training data of every single form. On the other hand, inflection offers us some potential advantages: by analyzing a word's morphology, it is possible to obtain information about its part of speech, e.g. in English a suffix of -ed or -ing might indicate a verb. A full morphological analysis involves determining a word's lemma (the canonical form) and its inflection. Because in German inflection is mostly (but not always) agglutinative, a word's last letters often carry the inflectional information. For example in the word *Kindern*, [Kind] is the lemma, [er] contains the plural, and [n] the dative part.

The German POS tagger in JET does not perform a real morphological analysis, but it keeps statistics of fixed-length affixes. For every word that is longer than this fixed length L, its first, respectively last L characters are considered the pre-, respectively suffix. The emission probability of a word is estimated as the product of the relative frequency of the word and the relative frequency of its affixes. We found that looking at prefixes of length 6 and suffixes of length 3 gives the best results (see table 2.2). Considering just suffixes results in a larger improvement of the performance than just considering prefixes. This is expected, as German inflection mainly concerns suffixes. But we show that also keeping statistics over prefixes improves the performance of POS tagging (see section 2.5). This can be explained by the fact that some prefixes in German are typical for certain parts of speech, such as *ent- (entkommen, entfernen, entstehen,...)* or *ver- (verlieren, verachten, verschlafen,...)* for verbs, and *Un- (Unfall, Unachtsamkeit, Unglaube,...)* for nouns. Finding the ideal length for the pre- and suffix was done experimentally. If the affixes are too short, the relative frequencies do not vary a lot among the different parts of speech, and if they are too long they become essentially the whole word, which also reduces their usefulness. Of course it would also be possible to keep statistics of affixes of variable length. Because the solution with fixed-length affixes proved to be effective already, this was not attempted.

2.4.3 Tag conversion

As mentioned above, the STTS tag set contains 54 tag types. We found that by combining some of these tag types, e.g. combining all 12 verb types into one, we achieve better performance on the resulting tag set. The simplified tag set consists of 23 part of speech types (see Appendix A). Performing this conversion on output (i.e. the POS tagger is trained on the original tag set and just outputs the corresponding simpler tags while tagging) results in slightly better performance than performing it before training (i.e. the POS tagger is trained on the simpler tag set), see table 2.4. Even though the performance on this simpler tag set is better, chunking (see chapter 3) performs better on the original tag set.

2.5 Results

We evaluated the POS tagger on the TüBa-D/Z corpus. The corpus consists of ca. 65,000 annotated sentences. Table 2.1 shows the performance of the tagger depending on the number of sentences used for training, with the rest of the corpus used for evaluation respectively. We see that we are close to the maximum performance when training with 32.000 sentences (or ca. 50% of the corpus) already. Also the percentage of unknown words remains relatively stable around 8% with a training set larger than half of the corpus. The following results are all based on a training set of 55.000 sentences (84% of the corpus).

The results in this section are based on the tokenization as provided by the treebank. When performing tokenization using the JET tokenizer we observe a misalignment of 0.19% of the tokens.

Table 2.2 shows how the pre- and suffix length that is taken into consideration affects the accuracy. We see that generally, suffixes are more important, but keeping statistics about prefixes increases the performance compared to the best performing configuration with just suffix statistics. Keeping no statistics about affixes results in an accuracy of 95.07%. Keeping just statistics about suffixes of length 3 we can increase the accuracy to 95.66%, while the best performing prefix

Sentences used for training	**Percentage of corpus**	**Accuracy (%)**	**Unknown words (%)**
1.000	1.5	89.5	30.4
2.000	3.1	91.7	24.9
4.000	6.1	92.9	20.5
8.000	12	94.1	16.6
16.000	24	94.7	13.4
32.000	49	95.4	10.5
48.000	73	95.7	8.8
55.000	84	95.5	8.7
60.000	91	95.7	8.3

Table 2.1: Performance of the POS tagger depending on training set size

length 4 only increases the accuracy to 95.32%. The best combination of pre- and suffixes, 6 and 3, respectively, results in an accuracy of 95.77%.

	Prefix length							
Suffix length	0	1	2	3	4	5	6	7
0	95.07	94.69	94.87	95.25	**95.32**	95.3	95.29	95.21
1	94.74	94.32	94.52	95.0	95.04	95.0	94.99	94.92
2	95.15	94.88	94.4	95.21	95.35	95.31	95.33	95.27
3	**95.66**	95.2	95.32	95.65	95.72	95.73	**95.77**	95.73
4	95.65	95.25	95.3	95.61	95.67	95.65	95.69	95.68
5	95.62	95.25	95.28	95.6	95.62	95.59	95.65	95.63
6	95.48	95.17	95.19	95.52	95.56	95.53	95.52	95.5
7	95.36	95.04	95.1	95.44	95.47	95.45	95.45	95.4

Table 2.2: Influence of prefix and suffix length on accuracy (using full STTS tagset)

In table 2.3 we see the performance gained from keeping statistics about affixes, compared to a basic HMM and a HMM that just keeps statistics about word features for the estimation of unknown word probabilities, as used in JET's English HMM tagger.

Finally, in table 2.4 we compare the best performing configuration of the JET part of speech tagger to the published performance of state-of-the-art systems for German POS tagging.

System	Accuracy (%)
Basic HMM	90.16
HMM + word features	91.81
HMM + word features + affix statistics	95.77

Table 2.3: Increased accuracy by keeping statistics about word features and affixes

System	Accuracy (%)
TnT	96.70
TreeTagger	97.53
JET with STTS	95.77
JET trained on full STTS, outputting the simple tag set	97.70
JET trained on simple tag set	97.03

Table 2.4: Comparison to state-of-the-art systems (numbers for systems other than JET from [3])

Chapter 3

Chunking

After tagging the words of a document with their parts of speech, the next step is assigning a syntactic structure to the sentences. A sentence can be represented as a syntactic tree, and identifying this tree is called parsing. Parsing is a complicated problem, and for many language processing tasks, including ours, full parsing is not necessary. A less complex, faster alternative is identifying flat, non-overlapping and non-recursive segments of the sentence that contain the parts of speech that we're most interested in, such as nouns or verbs, as a head (i.e. rightmost word). This process is called chunking, and the segments are called chunks. Because the chunks are not overlapping, they can easily be illustrated using a bracketing notation, such as:

$[_{NC}$The morning flight] $[_{PC}$from] $[_{NC}$Denver] $[_{VC}$has arrived.]

Here, NC stands for noun chunk, PC for prepositional chunk and VC for verb chunk. Note that, because chunks are non-overlapping, these chunks are not necessarily whole phrases: the prepositional phrase in this sentence for example would include *Denver*. This example, as well as a more in-depth explanation of chunking can be found in [9].

3.1 The JET chunking component

The JET chunker is an IOB tagger. IOB tagging is a standard approach for chunking that views the chunking process as a sequential classification, similar to part of speech tagging [9][14]. In IOB tagging, every word is tagged with one tag: either as a beginning (B) or an internal (I) part of a chunk, or as being outside (O) of any chunk. For the German chunking task, we extended the JET chunker with the ability to tag more than one category of chunks. Chunks are then tagged as B-X or I-X, where X is the chunk type.

The chunker finds the most likely sequence of IOB tags using a maximum entropy model, based on the open source Apache OpenNLP library's GISModel class [13].

3.2 German chunks

In general, chunking can be applied to German. There is one non-trivial problem that does not exist in English, though: In German, pre-head modifiers in noun phrases can contain noun and prepositional phrases. This means there is a recursive structure that cannot be represented by chunks. The following excerpt provides an example (from [10]):

$[_{NC}$der $[_{NC}$seinen Sohn$]$ liebende Vater$]$

Kübler et al. [10] propose a solution: the embedding noun chunk is split up and a new type of chunk is introduced, the 'stranded noun chunk' (sNC), so that the excerpt above would be tagged as:

$[_{sNC}$der$]$ $[_{NC}$seinen Sohn$]$ $[_{NC}$liebende Vater$]$

This way we again have a non-overlapping chunk structure. The authors of the before mentioned paper provided us with a version of the TüBa-D/Z corpus where the syntactic trees are converted into a chunk structure according to these rules.

3.3 Tag conversion

The TüBa-D/Z based chunk corpus from Kübler et al. marks 18 different chunk types: besides noun chunks also finite and infinite verb chunks, adjectival and adverbial chunks and others. Additionally some of these chunk types have sub types for named entities such as persons, locations or organizations. Prepositional phrases are marked as one single chunk, such as in this example:

$[_{PC}$In einer anonymen Anzeige$]$ $[_{VCFIN}$werden$]$ $[_{NC}$der Bremer Staatsanwaltschaft$]$ $[_{NC}$Details$][_{PC}$über dubiose finanzielle Transaktionen$]$ $[_{VCFIN}$mitgeteilt$]$.

For our purposes, we are mainly interested in identifying noun chunks. It is therefore more useful to identify the noun phrases inside prepositional phrases as noun chunks than tagging the whole prepositional phrase as a prepositional chunk. We transformed the chunk corpus, so that the above sentence is represented as such:

[$_{PC}$In] [$_{NC}$einer anonymen Anzeige] [$_{VCFIN}$werden] [$_{NC}$der Bremer Staatsanwaltschaft] [$_{NC}$Details][$_{PC}$über] [$_{NC}$dubiose finanzielle Transaktionen] [$_{VCFIN}$mitgeteilt].

As our goal is just identifying noun chunks, we have two possibilities: either we train the chunker on just the noun chunks, or we train the chunker on more chunk categories, but only let it output noun chunks. We tested four different training sets where the following chunk categories were annotated (listed in decreasing number of chunk categories):

- All chunk types, including the subtypes for named entities
- All chunk types, not including the subtypes for named entities (these were merged into their parent category, i.e. mainly NC)
- Just noun, verb, and "other" chunk type (the different verb chunk categories were merged into one, and all other categories were merged into the category "other")
- Just noun chunk type (all other chunk categories were treated as outside of any chunk)

For all of these configurations, the output of the chunker was just the noun chunks. The model with the most chunk categories performed best (see section 3.4). We therefore observe that even if we care about only one chunk category, it is worth training on a more complicated model.

3.4 Results

The chunker was evaluated on the TüBa-D/Z corpus using the evaluation script for the CoNLL-2000 shared task [18]. We only discuss the performance on noun chunks in this section, as we are not interested in other types of chunks for our task at hand.

We used 15.000 sentences for training. The OpenNLP GISModel class [13] needs memory proportional to the size of the training corpus to generate the maximum entropy model, making it not feasible to train it on a larger part of the corpus. But we found that the performance does not increase by a large margin any more when using a larger training set than 10.000 sentences (see table 3.1).

In table 3.2 we present the dependency of the chunking performance on the part of speech tags used by the chunker. We see that despite the better POS

Sentences used for training	Precision	Recall	F-measure
10	67.70%	74.94%	71.14
100	88.04%	90.33%	89.17
1.000	92.55%	92.79%	92.67
5.000	94.23%	94.27%	94.25
10.000	94.62%	94.64%	94.63
15.000	94.87%	94.82%	94.85

Table 3.1: Performance of the chunker on noun chunks with different sizes of training sets (using perfect part of speech tags)

POS tags	POS accuracy	Precision	Recall	F-measure
perfect POS tags from corpus	100%	94.87%	94.82%	94.85
POS tags from tagger, full STTS tag set	95.77%	91.44%	91.86%	91.65
POS tags from tagger, simple tag set	97.70%	89.32%	90.19%	89.75

Table 3.2: Performance of the chunker on noun chunks using different part of speech tags

accuracy when using the simple tag set, the chunker performs better on the full STTS tag set.

Finally, in table 3.3 we present the performance of the chunker on noun chunks using different training sets. We see that training the chunker on all chunk tags available in the corpus results in the best performance on noun chunks.

Trained on	Precision	Recall	F-measure
all chunk tags	91.44%	91.86%	91.65
no named entities	90.71%	91.03%	90.87
noun/verb/other	90.37%	90.83%	90.60
only noun chunks	90.51%	90.61%	90.56

Table 3.3: Performance of the chunker on noun chunks with different chunk training sets (using POS tags from the tagger, full STTS tag set)

Chapter 4

Patents

Our training data for part of speech tagging and noun chunking is newspaper text and in the previous chapters the evaluation was done with data from the same corpus. When training and test data is very similar, there is always a danger that overfitting gets rewarded [3]. As the final objective of this project is to extract information from patent documents, we have to compare the language used in patents with the language used in our training corpus. We were provided with a corpus of 500 patents in XML format for testing. We do not have German patents that are annotated with part of speech tags, therefore we can not report the accuracy of the POS tagger on German patents. We did annotate a patent with 531 noun chunks though and got a precision of 84% and recall of 88% (F-measure: 86%). Compared to the newspaper text, there is a higher number of unknown words: while testing on the TüBa-D/Z corpus results in ca. 8% of unknown words, patents have an average rate of unknown words >20%. A high number of these unknown words are compound nouns, not surprising given the bureaucratic language in patents. Another stumbling block for the newspaper-trained tools is the high frequency of numbers occurring in patents: not only are paragraphs numbered in many patents, but numbers also refer to claims and figures, and constructs such as 'DE 31 35 043 C2' refer to other patents. Often, numbers in parentheses are appended to nearly every noun, if they refer to certain parts of an attached drawing.

4.1 Processing of patent documents

We perform the following actions on a patent document:

1. The relevant portions of the patent are extracted using XPath and each saved into their own JET Document object. A patent consists of a title, an abstract, a description and a list of claims. Abstracts in German are not

present in every document, but there is an English abstract and title for every patent. Currently we do not make use of the English texts, but in the future it is possible that a cross-validation between the English and German components could be made using the English abstract. The XML files also contain a lot of information that we don't consider at the moment, such as a list of inventors and right holders, legal events, etc. By storing each section of the patent in its own Document object, it is possible to perform different actions or assign different weights to each section. Currently all sections are treated equally though.

2. Part of speech tagging is performed on each section of the patent.

3. Noun chunks are identified.

4. Terminology is extracted (see chapter 5)

5. Hearst patterns are extracted (see chapter 6)

Chapter 5

Terminology extraction

Terminology extraction is the task of identifying terms in a document that are specific to the document's domain. In English, this is a difficult problem, because terminology often consists of multi-word terms. We found that for our task at hand - extracting terminology from German patents - identifying single-word terms already brings promising results. Hong et al. [8] find that depending on the domain, from 57% and up to 94% of all terminology is single-word terms. Our very simple method for detecting terminology in patents works as follows:

1. For each word tagged as noun in the patent, check if it occurred in the training corpus of newspaper articles. If not, it is a candidate for being terminology.

2. If a candidate for terminology occurs a second time in the same patent, it is considered terminology.

The reason that terminology in German tends to consist of only one word is the formation of compound nouns. While English compound nouns are more often than not separated by a space, German grammar allows combining arbitrary nouns into new compound constructs. Our observations suggest that in patents new compound nouns are formed even more than in average German texts.

In the corpus of 500 patents, this method extracts on average about 150 words as candidates and 50 words as terminology, of which almost all are compound nouns. This number might be a bit high, i.e. not all of these terms would actually be considered terminology by a rigid definition. But considering the simplicity of the method, the results are promising.

Chapter 6

Hearst patterns

One particular semantic relation which we are interested in extracting is the EXEMPLIFY relation: whether one term is a subtype or instance of another term. Marti Hearst was one of the first to describe how such hyponymy relations might be extracted from (English) text using a set of patterns, now referred to as Hearst patterns [7]. The idea is simply to look for certain strings that often indicate a hyponym/hypernym relation. If there is a noun phrase to the left and to the right, it is assumed that these two noun phrases are in such a relation to each other. We implemented a corresponding set of German patterns as a demonstration of how our linguistic analysis (POS tagging and chunking) could be used. The following patterns are considered:

English	German
NP *as for example* NP	NP *wie zum Beispiel* NP
NP *is a* NP	NP *ist ein[e]* NP
NP *such as* NP	NP *wie etwa* NP
NP *including* NP	NP *einschließlich* NP
NP *or other* NP	NP *und andere* NP
NP *and other* NP	NP *oder andere* NP

Table 6.1: Hearst Patterns (from [5])

The drawback of the simplicity is that there is a high degree of false positives; [5] shows a precision of just 7.7% for Hearst patterns in German encyclopedic texts. In the corpus of 500 patents, we match in average 2.7 Hearst patterns per document. The patterns are filtered, keeping only pairs where either the hyponym or hypernym noun phrase contains a word considered terminology (see 5). There are on average 1.4 such pairs per document. The precision is about the same as shown in [5] - below 10%. Errors are mostly due to these reasons:

- There are many grammatical constructs that contain the patterns, but do not indicate a hyponym/hypernym construct. Especially *ist ein(e)* occurs often with another meaning, for example with *ist* taking the role of an auxiliary verb: "An der der Vorderseite abgewandten Seite des Abschnitts des erfindungsgemäßen Abdichtungsprofilstrangs *ist eine* Klebschicht aufgetragen" ("*an* adhesive layer *is* applied").
- Many patents contain references to numbered figures accompanying the document. These references are often in the form of *(Fig.) X ist ein Y* (*(fig.) X is a Y*), where X is the number of the figure and Y a description. These pairs are of course not useful hyponym/hypernym pairs. But on the other hand, it could be useful to identify the entities that are pictured in the figures, as they are likely to be important parts of the patent.

Chunking is not a big source of errors - most parts of the Hearst patterns are correctly identified as noun chunks.

Chapter 7

Conclusion and Future work

We have presented how some components of the JET system were adapted to German. The next step will be to merge the adapted system with the original English based components. It remains to be seen if some adaptations done for German will also result in improved performance for English.

As discussed in chapter 5, terminology extraction in German texts provides some promising advantages as compared to English. The agglutinative nature of German tends to result in more single word terms [8]. Single word terminology is easier to extract than bi- or multinomial terms, as frequency analysis becomes more straight-forward. A single word appearing only in one text or subject domain is highly likely to be terminology, while for a multinomial term more features have to be taken in consideration to decide if it is a candidate for terminology. It is conceivable that back-translation to English could result in improved performance even in English terminology extraction. This could be done either with parallel, directly translated English and German texts, but also for parallel related document groups, where the documents need not be exact translations.

Bibliography

[1] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, 2002.

[2] T. Brants, R. Hendriks, S. Kramp, B. Krenn, C. Preis, W. Skut, and H. Uszkoreit. Das NEGRA-Annotationsschema. *Negra project report, Universität des Saarlandes, Computerlinguistik, Saarbrücken, Germany*, 1997.

[3] E. Giesbrecht and S. Evert. Is part-of-speech tagging a solved task? An evaluation of POS taggers for the German web as corpus. In *Web as Corpus Workshop (WAC5)*, page 27, 2009.

[4] G. Görz and D. Paulus. A finite state approach to German verb morphology. In *Proceedings of the 12th conference on Computational linguistics-Volume 1*, pages 212–215. Association for Computational Linguistics, 1988.

[5] M. Granitzer, A. Augustin, W. Kienreich, and V. Sabol. Taxonomy extraction from German encyclopedic texts. In *Proceedings of the Malaysian Joint Conference on Artificial Intelligence*, 2009.

[6] R. Grishman. Tipster text architecture design. *New York University. www-nlpir. nist. gov/related_projects/tipster/docs/arch31. doc*, 1998.

[7] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

[8] Munpyo Hong, Sisay Fissaha, and Johann Haller. Hybrid filtering for extraction of term candidates from German technical texts. *In proceedings of terminologie et intelligence artificielle, TIA'2001*, 2001.

[9] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Prentice Hall, Upper Saddle River, N.J., 2nd ed edition, 2009.

[10] S. Kübler, K. Beck, E. Hinrichs, and H. Telljohann. Chunking German: an unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151. Association for Computational Linguistics, 2010.

[11] Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 111–119, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[12] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

[13] Tom Morton and Jason Baldridge. Class GISModel. http://maxent.sourceforge.net/api/opennlp/maxent/GISModel.html.

[14] L.A. Ramshaw and M.P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94. Cambridge MA, USA, 1995.

[15] I. Rehbein and J. Van Genabith. Why is it so difficult to compare treebanks? TIGER and TüBa-D/Z revisited. 2007.

[16] A. Schiller, S. Teufel, C. Stöckert, and C. Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS. In *Draft Universitat Stuttgart Institut fur maschinelle SprachverarbeitungUniversitat Tubingen Seminar fur Sprachwissenschaft*, 1999.

[17] H. Telljohann, E.W. Hinrichs, S. Kübler, H. Zinsmeister, and K. Beck. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). In *Seminar für Sprachwissenschaft, Universität Tübingen, Germany*, 2003.

[18] E.F. Tjong Kim Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132. Association for Computational Linguistics, 2000.

Appendix A

The STTS tag set

The following table shows the 54 part of speech types in the STTS tag set. For our simplified tag set we merged similar tags, as indicated with the horizontal lines, resulting in a set of size 23.

	German description	**English description**	**examples**
ADJA	attributives Adjektiv	attributive adjective	[das] große [Haus]
ADJD	adverbiales oder prädikatives Adjektiv	adverbial or predicative adjective	[er fährt] schnell, [er ist] schnell
ADV	Adverb	adverb	schon, bald, doch
APPR	Präposition; Zirkumposition links	preposition; left circumposition	in [der Stadt], ohne [mich]
APPRART	Präposition mit Artikel	preposition + article	im [Haus], zur [Sache]
APPO	Postposition	postposition	[ihm] zufolge, [der Sache] wegen
APZR	Zirkumposition rechts	right circumposition	[von jetzt] an
ART	bestimmter oder unbestimmter Artikel	definite or indefinite article	der, die, das, ein, eine
CARD	Kardinalzahl	cardinal number	zwei [Männer], [im Jahre] 1994
FM	Fremdsprachliches Material	foreign language material	[Er hat das mit] A big fish [" übersetzt]
ITJ	Interjektion	interjection	mhm, ach, tja
KOUI	unterordnende Konjunktion mit *zu* und Infinitiv	subordinating conjunction with *zu* + Infinitive	um [zu leben], anstatt [zu fragen]
KOUS	unterordnende Konjunktion mit Satz	subordinating conjunction with clause	weil, daß, damit, wenn, ob
KON	nebenordnende Konjunktion	coordinative conjunction	und, oder, aber
KOKOM	Vergleichskonjunktion	particle of comparison, no clause	als, wie
NN	normales Nomen	noun	Tisch, Herr, [das] Reisen
NE	Eigennamen	proper noun	Hans, Hamburg, HSV

Table A.1: The STTS tag set (part 1), from [17]

	German description	English description	examples
PDS	substituierendes Demonstrativpronomen	substituting demonstrative pronoun	dieser, jener
PDAT	attribuierendes Demonstrativpronomen	attributive demonstrative pronoun	jener [Mensch]
PIS	substituierendes Indefinitpronomen	substituting indefinite pronoun	keiner, viele, man, niemand
PIAT	attribuierendes Indefinitpronomen ohne Determiner	attributive indefinite pronoun without determiner	kein [Mensch], irgendein [Glas]
PIDAT	attribuierendes Indefinitpronomen mit Determiner	attributive indefinite pronoun with determiner	[ein] wenig [Wasser], [die] beiden [Brüder]
PPER	irreflexives Personalpronomen	irreflexive personal pronoun	ich, er, ihm, mich, dir
PPOSS	substituierendes Possessivpronomen	substituting possessive pronoun	meins, deiner
PPOSAT	attribuierendes Possessivpronomen	attributive posessive pronoun	mein [Buch], deine [Mutter]
PRELS	substituierendes Relativpronomen	substituting relative pronoun	[der Hund ,] der
PRELAT	attribuierendes Relativpronomen	attributive relative pronoun	[der Mann ,] dessen [Hund]
PRF	reflexives Personalpronomen	reflexive personal pronoun	sich, einander, dich, mir
PWS	substituierendes Interrogativpronomen	substituting interrogative pronoun	wer, was
PWAT	attribuierendes Interrogativpronomen	attributive interrogative pronoun	welche [Farbe], wessen [Hut]
PWAV	adverbiales Interrogativ- oder Relativpronomen	adverbial interrogative or relative pronoun	warum, wo, wann, worüber, wobei
PROP	Pronominaladverb	pronominal adverb	dafür, dabei, deswegen, trotzdem
PTKZU	*zu* vor Infinitiv	*zu* + infinitive	zu [gehen]
PTKNEG	Negationspartikel	negation particle	nicht
PTKVZ	abgetrennter Verbzusatz	separated verb particle	[er kommt] an, [er fährt] rad
PTKANT	Antwortpartikel	answer particle	ja, nein, danke, bitte
PTKA	Partikel bei Adjektiv oder Adverb	particle with adjective or adverb	am [schönsten], zu [schnell]
TRUNC	Kompositions-Erstglied	truncated word - first part	An- [und Abreise]
VVFIN	finites Verb, voll	finite main verb	[du] gehst, [wir] kommen [an]
VVIMP	Imperativ, voll	imperative, main verb	komm [!]
VVINF	Infinitiv, voll	infinitive, main	gehen, ankommen
VVIZU	Infinitiv mit *zu*, voll	infinitive + *zu*, main	anzukommen, loszulassen
VVPP	Partizip Perfekt, voll	past participle, main	gegangen, angekommen
VAFIN	finites Verb, aux	finite verb, aux	[du] bist, [wir] werden
VAIMP	Imperativ, aux	imperative, aux	sei [ruhig !]
VAINF	Infinitiv, aux	infinitive, aux	werden, sein
VAPP	Partizip Perfekt, aux	past participle, aux	gewesen
VMFIN	finites Verb, modal	finite verb, modal	dürfen
VMINF	Infinitiv, modal	infinitive, modal	wollen
VMPP	Partizip Perfekt, modal	past participle, modal	gekonnt, [er hat gehen] können
XY	Nichtwort, Sonderzeichen enthaltend	non-word containing special characters	3:7, H2O, D2XW3
$,	Komma	comma	,
$.	Satzbeendende Interpunktion	sentence-final punctuation	. ? ! ; :
$(	sonstige Satzzeichen; satzintern	other sentence internal punctuation	- [,]()

Table A.2: The STTS tag set (part 2), from [17]

Appendix B

Example of terminology and hearst patterns extracted from a patent

B.1 Patent text (excerpt)

Gegenstand der Erfindung ist ein System mit einer Mehrzahl von Schlössern (2), deren Riegel jeweils mittels eines Schlüssels oder elektromotorisch ver- oder entriegelbar sind, wobei bei Bewegung des Riegels in die Schließstellung mittels eines schlüsselbetätigten Schließzylinders zuvor die Riegel der anderen Schlösser motorisch verriegelt werden. Nach dem Verschließen aller zum System gehörigen Schlösser wird eine Einbruchmeldeanlage scharf geschaltet. Die Schlösser (2) und/oder Verteiler, von denen Leitungen zu Schlössern und/oder Gebern (9, 10) und/oder von Hand betätigbaren Tastern (12) bzw. Schaltern verlaufen, sind mittels einer wenigstens einen Prozessor aufweisenden Schaltung als Teilnehmer an einen gemeinsamen Bus (16) eines Netzwerks ausgebildet.

B.2 Terminology (excerpt)

S10 Endschalter Übertragungsleitung Taster Alarmmeldung Prozessor Geber Teilnehmerschaltung Blinken Übertragung Gebers Reed-Kontakt Schaltzustand Relais SteuereinheitPlatine Motorblockschloß Tastern Bauelemente S01 Aktoren Schließstellung DIP-Schalters Spule Hierdurch Profilzylinderkerns Riegels Schließzylinderkerne Baugruppe Steuereinheiten Zeitabständen Zufallszahl Abfrage Autorisierung Sy DIP-Schaltern Blockschlosses DIP-Schalter Kom Türblatt Gebäudebereichen Einbruchmeldeanlage Türraum Schließnasen Steckverbinder Informationsübertragung V-Netzes Autorisierungs-Routine Schließzylinder Programmiergerät Schlosses Koppelelemente Spannungsversorgung Betriebsspannung Teilnehmerschaltungen Detektoren Informationssignale EMA Profilzylinderkerne Schließnase Identifiziernummer Elektromagneten Schloßriegel Verriegelungsstellung Türverteilern Teilnehmers Personal-Computer Identnummer Sabotagemeldung

B.3 Hearst patterns

jeden Gebäudebereich:eine eigene Steuereinheit (ist eine)
17:eine Steuereinheit (ist eine)
übertragbare hochfrequente Signale:der 220 V-Netzleitung (und andere)
diesem System:eine zentrale Steuereinheit (ist eine)
dieser Alternative:eine Steuereinheit (ist eine)
des einen oder anderen Profilzylinderkerns:des einen oder anderen Profilzylinderkerns (oder andere)
der Türzarge:ein Reed-Kontakt (ist ein)
jeden Gebäudebereich:eine eigene Steuereinheit (ist ein)
5:ein Geber (ist ein)
Bus:eine Steuereinheit (ist ein)
Bus:eine Steuereinheit (ist eine)
dieser Alternative:eine Steuereinheit (ist ein)
diesem System:eine zentrale Steuereinheit (ist ein)
17:eine Steuereinheit (ist ein)
24:ein Deckelkontakt (ist ein)